Card Management of the FIPS 201 Personal Identity Verification Card

Draft Version 1.0 – S1

October 16, 2004

Table of Contents

1. Introduction

Operations on a FIPS 201 Personal Identity Verification card that create, alter or delete critical security parameters or create or delete data structures are generally referred to as card management operations. Examples of card management operations the generation of a private key on the card or the creation of a new data file on the card

Typically card management operations are only performed by the card issue or an application provider although a cardholder changing their PIN is strictly speaking also an example of a card management operation. Card management includes both card personalization – making the card ready for use by a particular individual – and card administration – adding a application to the card.

Card management operations are differentiated from card use operations because they utilize commands, communication protocols and cryptographic algorithms that are not used during normal card usage. The ISO standard ISO/IEC 7816-9, entitled "Interindustry commands for card and file management" describes the ISO commands and procedures that are used for card management whereas the ISO standard ISO/IEC 7816-4, "Organization, security and commands for interchange" describes the ISO commands and procedures that are used in everyday use.

That said card management is not some sort of backdoor into the card operating system. Card management commands are accessed on the card edge in exactly the same way that card usage commands are accessed. The security of card management as with the overall security of the card itself is based on proper key management and key usage not on hidden functionality

Of course, card management and card use are intimately tied together. The purpose of card management is to insure that the security architecture implemented on the card faithfully reflects the security policy that the card is intended to implement and enforce.

This note describes how the client-application programming interface and the card platform commands of the FIPS 201 Personal Identity Verification card are used to perform card management. The discussion is driven primarily by a series of card management examples.

1.1 Prerequisites

The reader is assumed to be familiar with the FIPS 201 specification and optimally to have a copy at hand.

The reader is also assumed to be generally familiar with integrated circuit card security and the integrated circuit card security architecture as described in the ISO/IEC 7816 and ISO/IEC 24727 series of integrated circuit card standards. Of particular relevance are ISO/IEC 7816-9, "Interindustry commands for card and file management" and ISO/IEC 7816-13, "Commands for application management in multi-application environment".

2. Terms, Acronyms, and Notation

2.1 Terms

Card Application	Set of data elements and associated card command implementations that can be selected using an application identifier (AID).

Card Manager	The distinguished card application present on every PIV integrated circuit card. The Card Manager is the currently selected application when the card is powered up or reset.
Card Management	Operations on a card that create, alter, or delete critical security parameters on the card or add new data structures to or delete data structures from the card.
Client Application	A computer program running on a computer connected to an interface device containing an integrated circuit card and using the application programming interface described herein to access the capabilities of the integrated circuit card.
Cryptographic Information Application	A database on the FIPS 201 card that describes cryptographic material such as keys and certificates that are loaded on and known to the card.
Principal	An entity whose credentials can be authenticated using reference data and authentication protocols on the PIV.
Reference Data	Data used to perform an authentication protocol for a specific principal. Examples are passwords, PINs, and cryptographic keys used for authentication.

2.2 Acronyms

ADF	Application Dedicated File
AID	Application Identifier
BER	Basic Encoding Rules
DF	Dedicated File
EF	Elementary File
FCP	File Control Parameters
FID	File Identifier
ICC	Integrated Circuit Card
IFD	Interface Device
MF	Master File
PIN	Personal Identification Number
PIV	Personal Identity Verification
RFU	Reserved for Future Use
TF	Transparent File
TLV	Tag-Length-Value

2.3 Notation

All lengths are measured in number of bytes unless otherwise noted.

3. FIPS 201 Card Management Overview

The FIPS 201 Card Manager is a card application that is present on every FIPS 201 card and it is the on-card representative of the card issuer. A card management client-application is a client-application that interacts exclusively with the Card Manager. The Card Manager in turn interacts with the various

application domains on the card to perform card management operations. Figure 1 shows this relationship.

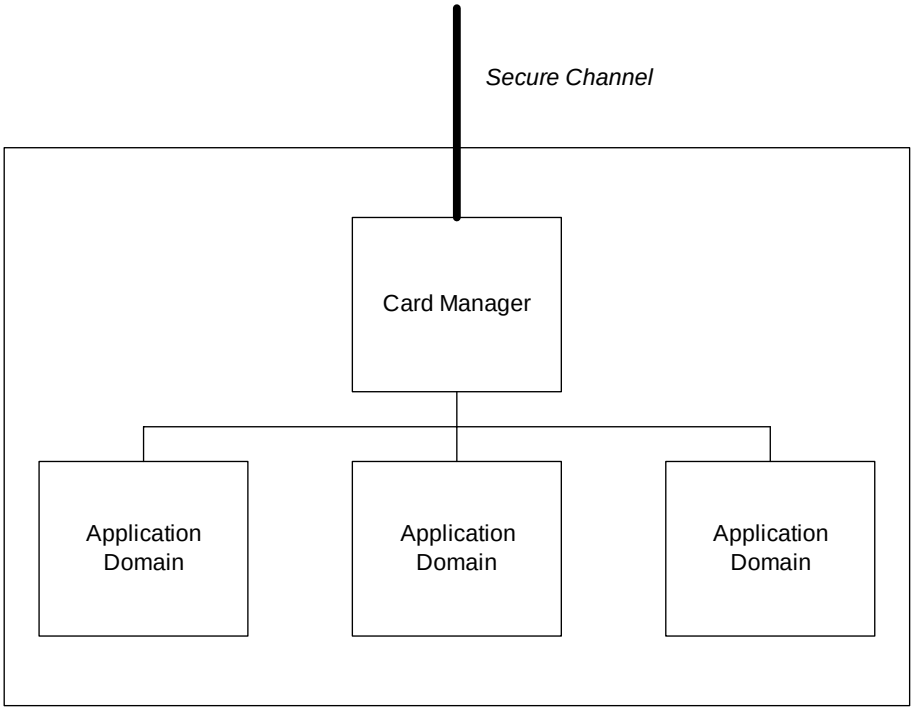

Figure 1: Card Manager and Application Domains

In interacting with the Card Manager, a card management client-application may use many of the same FIPS 201 client-application application programming interface entry points that a non-managment card application uses. A card management application, for example, has to establish communication with the card and often has to read and update existing files on the card.

These client-application programming interface entry points generate card-edge commands that are exactly the same card-edge commands generated by non-management applications with one important difference. Card management client-applications use a secure channel to communicate with the Card Manager.

3.1 Initiation of a Card Management Session

The beginning of a card management session from the point of view of calls the card management client-application places on the FIPS 201 Client-Application Programming Interface looks like this:

1. Establish communication with the card
   ```
   Connect
   Acquire Context
   ```
2. Select the Card Manager application
   ```
   Select Card Application
   ```
3. Establish a secure communication channel with the Card Manager
   ```
   Establish Secure Channel
   ```

3.2 Card Management Security

The Card Manager application (on behalf of the card issuer) is charged with insuring that a recognized principal is performing card management operations and furthermore that this recognized prinicipal is authorized to perform the card management functions being requested.

The Card Manager application accomplishes this by analyzing the cryptography applied to card management requests. Simply put, if the keys applied to a request are associated with a principal that is authorized to perform the request, then the request is granted. Otherwise, the request is rejected.

4. Examples of FIPS 201 Card Management

The following examples of FIPS 201 card management describe some typical card management functions. The descriptions are provided with respect to both of the interfaces defined in FIPS 201: the client-application programming interface and the card platform command interface.

A card management client-application would, of course, use only the client-application programming interface but it is instructive to see what is happening at the card-edge to get an end-to-end understanding of card management.

Since establishing a secure channel is a prerequisite for any card management function, this is operatin is described first.

4.1 Establishing a Secure Channel

The card management client-application establishes a secure channel with the Card Manager for two purposes. First, the cryptographic processes in a secure channel protect the information going to and coming back from the card from evesdropping and from alteration. Second, the particular keys used in the secure channel authenticate the card management client-application to the Card Manager and this authentication in turn authorizes the card management client-application to perform the card management operations it wishes to perform.

4.1.1 At the Client-Application Programming Interface

Figure 2 shows the client-application programming interface entry point used to establish a secure channel with a card to which a connection has been made is shown and on which the Card Manager is the currently selected application.

```
status_word EstablishSecureChannel(
      IN handle                    cardHandle,
      IN secure_channel_type       secureChannelType,
      IN reference_data_identifier transmissionKey,
      IN reference_data_identifier responseKey
)
```

Figure 2: API Entry Point to Establish a Secure Channel

The `cardHandle` argument is just the program identifier of the card to which the secure channel is to be established.

The `secureChannelType` argument describes the type of cryptographic processing that will be applied to the card management requests to be sent by the card management client-application. This description does not include the keys or particular cryptographic algorithms that will be used. This is left to the `transmissionKey` argument.

Rather `secureChannelType` describes whether or not the whole request will be encrypted, whether or not the request will be signed and whether or not a message authentication code will be added to the request. A card management client-application can pick one or two or all three of these and can use different keys for each type of processing. The choices of the card management application are communicated to the Card Manager

The `transmissionKey` argument describes the keys that are to be used for each of the cryptographic processes indicated in the `secureChannelType` argument. Note that this argument simply describes the keys to be used; it doesn't actually provide the keys.

The card management client-application and the Card Manager share a key name space that is stored in a database on the card called the *FIPS 201 cryptographic information application*. This database can be explored by the card management client-application to insure that the keys that it will ask the Card Manager to use are actually available on the card.

Finally, the `responseKey` argument tells the Card Manager what keys it is to use to send information back to the card manager client-application. As with the transmissionKey argument, the keys themselves are not provided but rather the names of the keys to be used as registered in the cryptographic information application database.

4.1.2 At the Card Platform Command Interface

A call on the `EstablishSecureChannel` entry point will not necessarily cause any activity on the card platform command interface.

If the client-application programming interface implementation ("middleware") has already explored the cryptographic information application on the card or implicitly knows what keys are available on the card based, for example, on the card's identification then the middleware can immediately accept or reject request to establish a secure channel based on this knowledge and what is in the request.

If the middleware does not know if the Card Manager has access to the keys described in the request to establish a secure channel, then the middleware will SELECT the cryptographic information application and use further SELECT commands and READ BINARY commands to explore the cryptographic information application database to insure that the requested keys are available on the card.

4.2 Creating a Data File

Creating a new data file typically entails two steps. First the file itself has to be created and second data has to be written into the file. In some cases the same keys that are being used for the secure channel can be used to create the file and to write data into it. In other cases, a different key may be needed to create the file and still a third key needed to write data into it.

4.2.1 At the Client-Application Programming Interface

An illustrative card management client-application would use the following calls to the client-application programming interface to create a new data file in a card application:

1. `SelectCardApplication` – to set the application in which the data file is to be created
2. `SelectDataElement` – one or more calls to this entry point to select the data element in which the new data file is to be created
3. `AuthenticatePrincipal` – to gain the privilege to create a data file in the currently selected data element
4. `CreateDataElement` – to create the new data file
5. `AuthenticatePrincipal` – to gain the privilege to write to the new data file
6. `WriteData` – to enter data into the new data file

Note that Steps 5 and 6 may not have to be executed if the authorization to create the file is sufficient to write data into the file.

4.2.2 At the Card Platform Command Interface

At the card platform command interface, these calls on the client-application programming interface could generate the following commands:

1. SELECT FILE BY AID – to select the application in which the new data file is to be created
2. SELECT FILE BY FID – to select the dedicate file (directory) in which the file is to be created
3. EXTERNAL AUTHENTICATE – to gain the privilege to create the file in the directory
4. CREATE FILE – to create the file
5. EXTERNAL AUTHENTICATE or VERIFY – to gain the privilege to enter data into the file
6. UPDATE BINARY – to write data into the file

If authorization to create the file is sufficient to write data into the file, then Steps 5 and 6 are not necessary. Content can be written into the file using the BER-TLV version of the CREATE FILE command with UPDATE BINARY in a command-to-execute TLV in the data file carrying the data.

4.2.3 An Alternative Method

The method for creating a new data file and loading data into described above uses a command, CREATE FILE, taken from ISO/IEC 7816-9. It is also possible to create a new file and load data into it using the card content management commands described in ISO/IEC 7816-13 and in the section below.

Which of these two approaches is used is a design decision for the software that implements the FIPS 201 client-application programming interface for it is this software that translates between calls on the entry points on this interface and commands sent over the card platform command interface.

4.3 Downloading an Application

Downloading a new application to the card is not different-in-kind from creating a new data file. A place has to be found to put the application and the data and code comprising the application have to be written into this place. In the case of an application there may be a final step of activating the application. This step is decoupled from the step of loading the application because it may useful to enable it to be taken at a time that is different from the time that the application is loaded onto the card.

4.3.1 At the Client-Application Programming Interface

4.3.2 At the Card Platform Command Interface

4.4 Injecting a Key

4.4.1 At the Client-Application Programming Interface

4.4.2 At the Card Platform Command Interface

4.5 Generating a Key

4.5.1 At the Client-Application Programming Interface

4.5.2 At the Card Platform Command Interface

5. References

5.1 ETSI

ETSI TS 102 221, *Smart cards; UICC-Terminal interface; Physical and logical characteristics*

ETSI TS 102 222, Integrated Circuit Cards (ICC); *Administrative commands for telecommunications applications*

5.2 FIPS

FIPS 140-2, *Security Requirements for Cryptographic Modules*, June, 2001

FIPS 180-2, *Secure Hash Standard (SHS)*, August, 2002.

FIPS 186-2, *Digital Signature Standard (DSS)*, January, 2000

FIPS 190, *Guideline for the Use of Advanced Authentication Technology Alternatives*, September, 1994

FIPS 196, *Entity Authentication Using Public Key Cryptography*, February, 1997

FIPS 197, *Advanced Encryption Standard*, November, 2001

FIPS 198, *The Keyed-Hash Message Authentication Code (HMAC),* March, 2002

5.3 ISO/IEC

ISO 7498-2:1989 Information Processing Systems - Open Systems Interconnection - Reference Model - Part 2: Security Architecture

ISO/IEC 7812-1:2000, *Identification cards — Identification of issuers — Part 1: Numbering system*

ISO/IEC 7816 (all parts), *Information technology — Identification cards — Integrated circuit(s) cards with contacts*

ISO/IEC 9796 (all parts), *Information technology — Security techniques — Digital signature schemes giving message recovery*

ISO/IEC 9797 (all parts), *Information technology — Security techniques — Message authentication codes (MACs)*

ISO/IEC 9798 (all parts), *Information technology — Security techniques — Entity authentication*

ISO/IEC 9979:1999, *Information technology — Security techniques — Procedures for the registration of cryptographic algorithms*

ISO 9992-2:1998, *Financial transaction cards — Messages between the integrated circuit card and the card accepting device — Part 2: Functions, messages (commands and responses), data elements and structures*

ISO/IEC 10116:1997, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*

ISO/IEC 10118 (all parts), *Information technology — Security techniques — Hash-functions*

ISO/IEC 10536 (all parts), *Information technology — Identification cards – Contactless integrated circuit(s) cards – Close-coupled cards*

ISO/IEC 11770 (all parts), *Information technology — Security techniques — Key management*

ISO/IEC 14443 (all parts), *Information technology — Identification cards – Contactless integrated circuit(s) cards – Proximity cards*

ISO/IEC 14888 (all parts), *Information technology — Security techniques — Digital signatures with appendix*

ISO/IEC 15693 (all parts), *Information technology — Identification cards – Contactless integrated circuit(s) cards – Vicinity cards*

ISO/IEC 18033 (all parts), *Information technology — Security techniques — Encryption algorithms*

ISO/IEC 24727-1 *Identification cards — Integrated circuit(s) cards programming interfaces — Architecture.*

ISO/IEC 24727-2 *Identification cards — Integrated circuit(s) cards programming interfaces — Generic card edge.*

ISO/IEC 24727-3 *Identification cards — Integrated circuit(s) cards programming interfaces — Programming interface.*

5.4 GlobalPlatform

GlobalPlatform Card Specification v2.1.1, March, 2003.

GlobalPlatform Load and Personalization Interface v1.0, March, 2003.

GlobalPlatform Guide to Common Personalization v1.0, March, 2003.